

Commandes utiles

Afficher l'état de la branche en cours :

```
git status
```

Ajouter des modifications à un commit :

```
git add <chemin vers mon fichier>
```

Commit :

```
git commit -m "Mon message de commit"
```

Envoie du commit :

```
git push
```

Récupération des derniers commits distants :

```
git pull
```

Supprimer le dernier commit :

```
git reset --hard HEAD~1
```

Options :

- hard
 - supprime le commit et enlève tes changements en local
- mixed
 - supprime le commit de la staging area mais les changements en local (retours à l'état nécessitant un `git add xxxx`).
- soft
 - supprime le commit mais reste dans la staging area (après un `git add xxxx`), plus qu'à re- commit !

Appliquer les commits d'une branche au dessus d'une autre branche:

```
git switch <branche cible>
```

```
git rebase <branche ou ajouter les commit>
```

Changer de branche :

```
git checkout <nom de ma branche>
```

Astuces :

Tu t'es trompé de branches et tu veux pas perdre ton code ?

Sauvegarder ton code dans un tampon :

Avant de stash, il faut avoir add ton code en faisant un `git add /path/to/file`

```
git stash
```

Fait ton changement de branche :

```
git checkout ....
```

Applique ton code en mémoire tampon :

```
git stash pop
```

Revision #4

Created 24 November 2024 16:36:56 by Nicolas

Updated 9 February 2025 14:11:07 by Nicolas